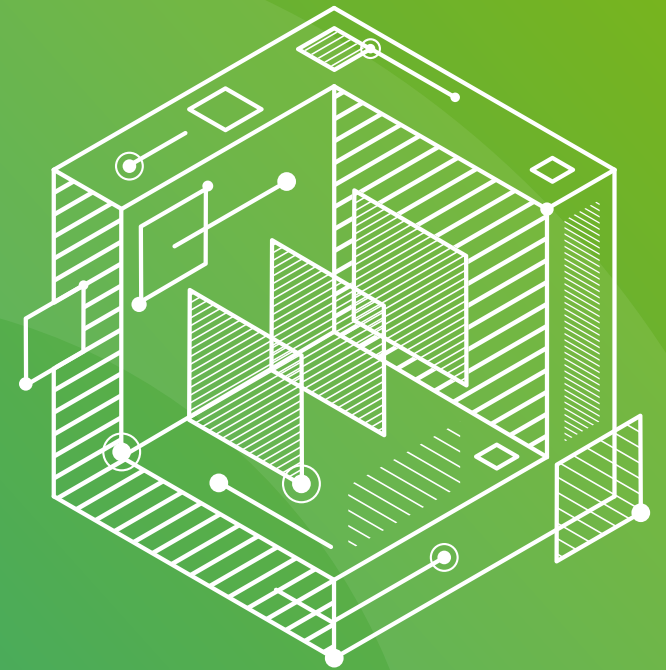
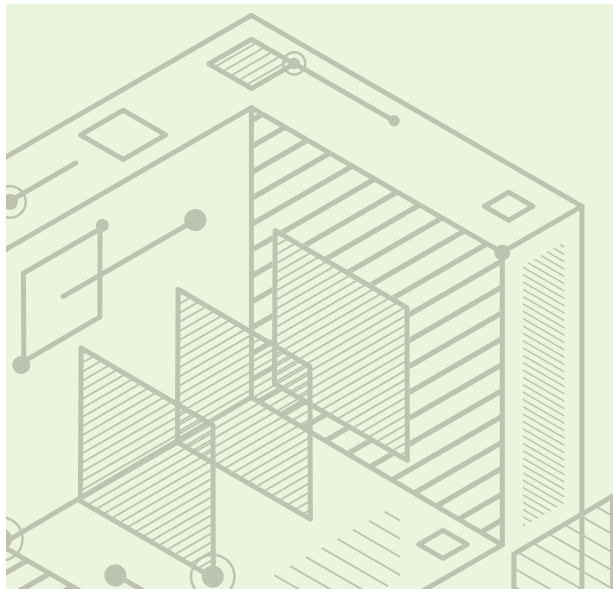


Prefetch

TECHNICAL CAPABILITY SHEET



open
DRIVES



Capability Definition

Prefetch is a specific type of data retrieval and caching function that Atlas Core provides by leveraging ZFS file system capabilities along with our own tuning and optimization code. Prefetch takes advantage of both the contextual awareness of the data being requested as well as the system's cache to fetch more data than is explicitly being requested. The purpose is to eliminate lag time and disruptive performance while retrieving additional information, because Atlas Core can read data from much faster system cache as opposed to reading it from slower storage media.

Note: The ZFS file system is not inherently tuned and optimized for video- and audio-based media. OpenDrives spent considerable R&D performing this tuning within Atlas Core specifically for large-file media, such as video files, so that working with this type of data is as fast as possible. Lastly, prefetch is not auto-tuning but rather is a manually tuned capability.

Outcome & Value

The outcome of the prefetch capability is enhanced performance through quicker data retrieval (from cache) and accelerated responsiveness as a result. By maintaining awareness of what the user/application is requesting and working with throughout a workflow or workload, Atlas Core anticipates the flow of requested data and retrieves a window of data around that explicit request in the prediction that that prefetched information, too, will be required. End users can expect little to no lag time within enterprise applications when accessing data.

How It Works

The prefetch capability is based on Atlas Core's ability to detect any linear access patterns, including forward or backward access. Once it detects an access pattern (contextual awareness), the prefetch capability then

leverages system cache to store subsequent data that is most likely to be requested around the currently accessed information. If the access pattern continues as anticipated, data is aggressively prestaged to much faster cache from slower storage media (such as spinning disk) in preparation of the data being requested. If the access pattern shifts to another portion of data, the cycle repeats with prefetching of information around the explicitly requested data.

Premiere-Specific Prefetch: With specific applications such as Adobe Premiere Pro, we make accessible an application-level cache that uses NVMe (with the Optimum, for example, we allocate 4 of the 8 NVMe SSDs) to provide faster data access for the prefetch capability. This area can store all cache files, working directories, and project files. This is a recommended best practice to accelerate Adobe Premiere Pro performance.

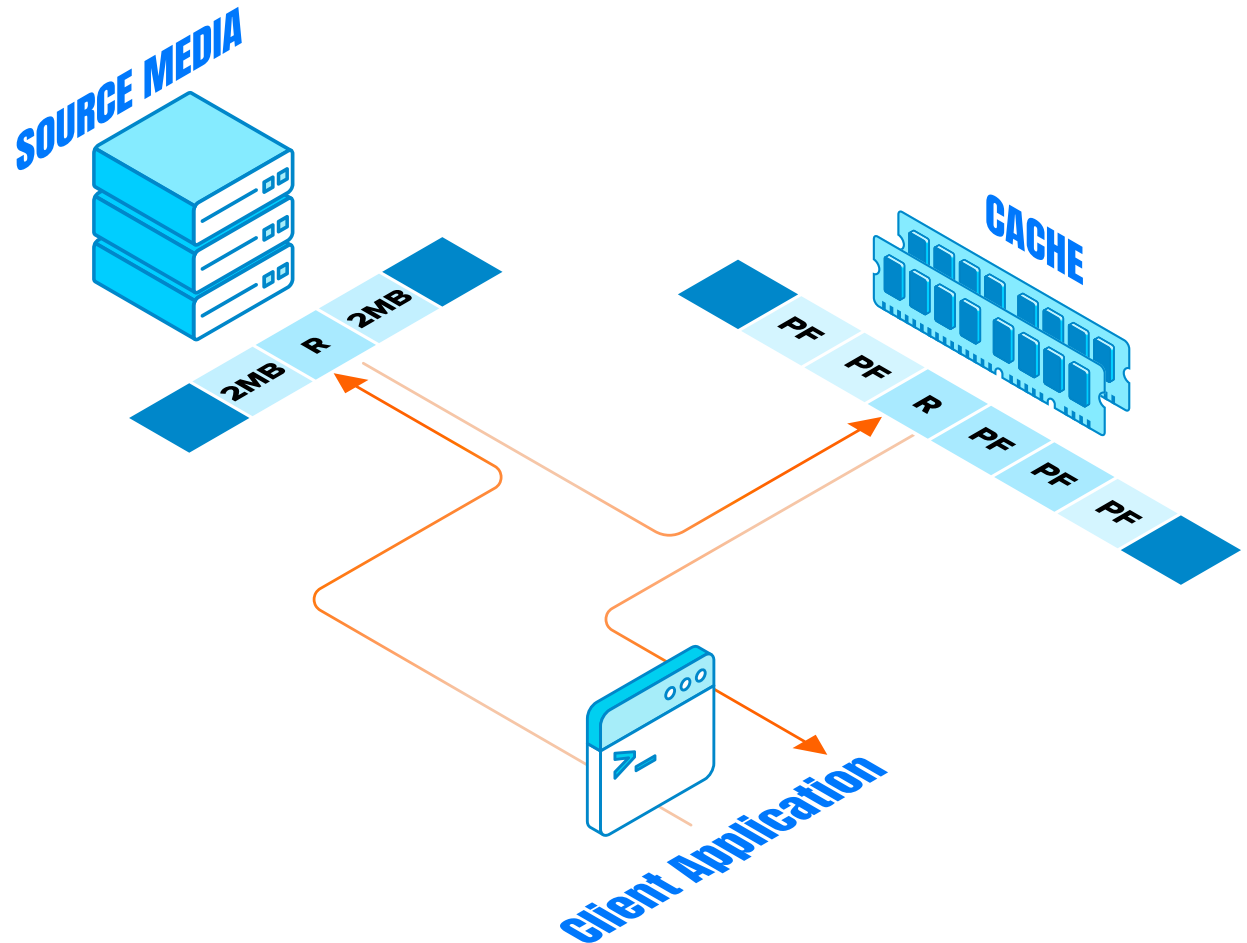
Characteristics

The prefetch capability has the following characteristics:

- Block-level (not file-level) caching
- Contextual awareness senses an access pattern, initiating prefetching data in a window around explicitly requested information—for example, contextual awareness derives whether information ahead of the current timestamp will likely be requested
- Window size from 4K to 2MB per fetch cycle (2MB by default)
- Up to 8 maximum prefetch streams per file
- Jumping around without establishing a clear access pattern deteriorates prefetch performance

The following diagram illustrates the prefetch process. The diagram displays a requesting application, the storage media, the requested information (R), the 2MB window around R, and prefetched data (PF).

When the user or application requests information stored on disk, Atlas Core/ZFS begins caching to memory the explicitly requested data and a 2MB window around it. As the access pattern becomes established, all data is read from the much faster cache rather than directly from storage.



Further Reading

As previously stated, prefetch is a type of predictive data caching within Atlas Core. For more information about caching in general, please refer to the Caching Capability Sheet which explains in more detail the mechanics behind our data caching capabilities which we've implemented within Atlas Core.